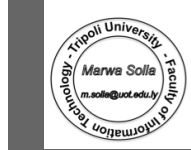
جامعة طرابلس ـ كلية تقنية المعلومات

Design and Analysis Algorithms

تصميم و تحليل خوارزميات

ITGS301

المحاضرة الرابعة: Lecture 4



خريف 2022

The Problem Of Sorting

Sorting

- Sorting is one of the most common data processing applications , the through which data are arranged according to their values.
- If data were not ordered , we would spend hours trying to find a single piece of information.

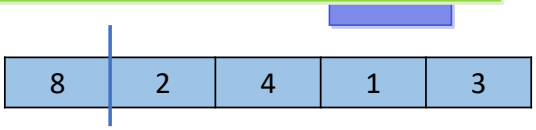
- Data may be sorted in either ascending sequence or descending sequence . and if the order of the sort is not specified its assumed to be ascending..

Insertion Sort

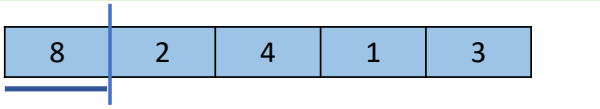
- In the insertion sort , the list is divided in two parts : sorted and unsorted.
- In each pass the first element of the unsorted sub list is transferred to the sorted sub list by inserting it at appropriate place.
- If we have a list of n elements , it will take at most n-1 passes to sort the data.

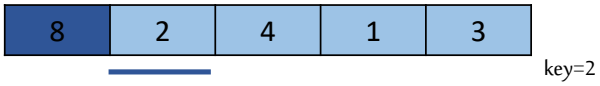
```
InsertionSort(A, n) {
    for i = 2 to n {
            key = A[i]
            j = i - 1;
            while (j > 0) and (A[j] > key) {
                    A[j+1] = A[j]
                    j = j – 1
            }
            A[j+1] = key

    }
}
```

```
InsertionSort( A, n)
{
For i = 2 to n {

   Key = A[i]
   j= i - 1

While ( j > 0) and (A[j] > key)
 {
  A[j+1] = A[j]
  j = j – 1
 }
 A[ j+1] = key
 }
```

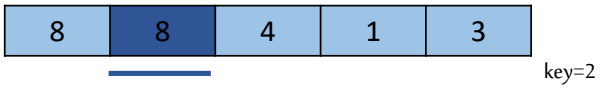


Example 1 :

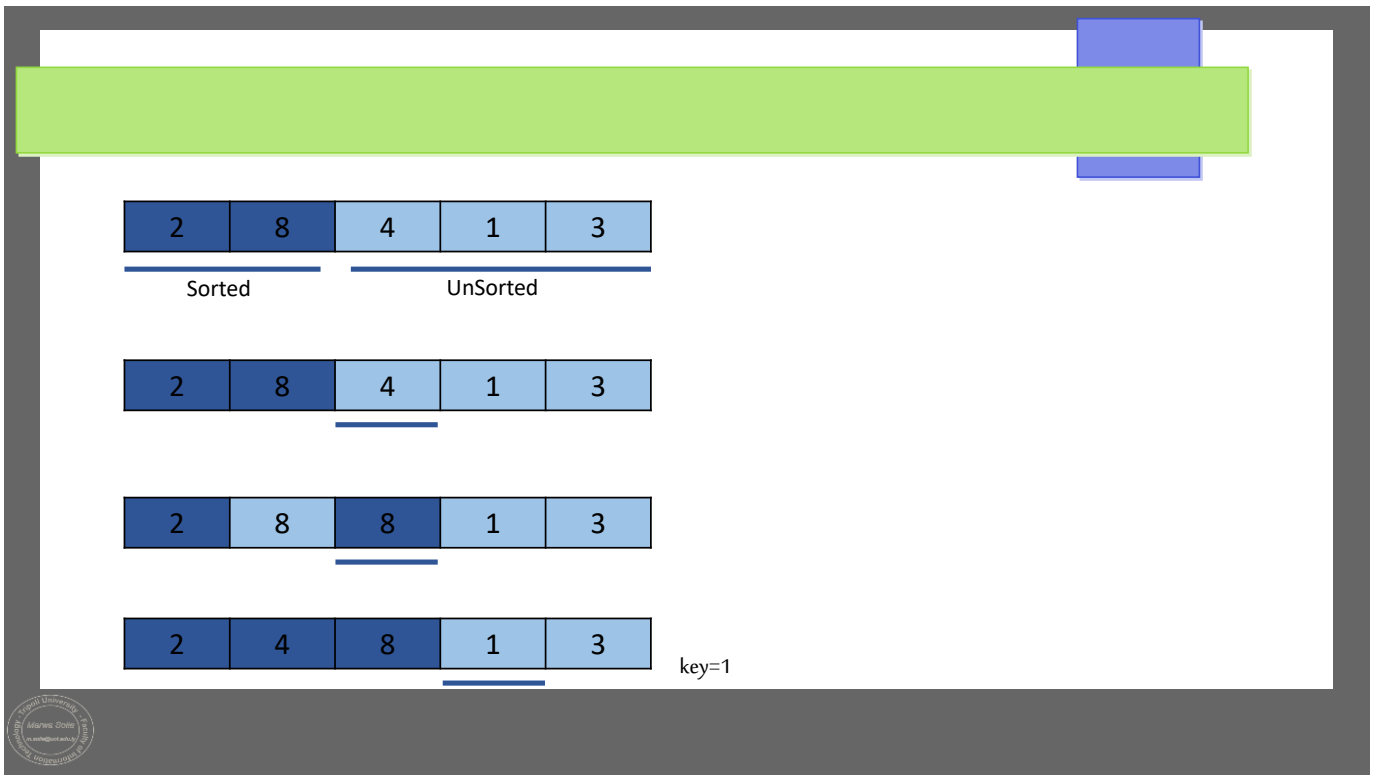


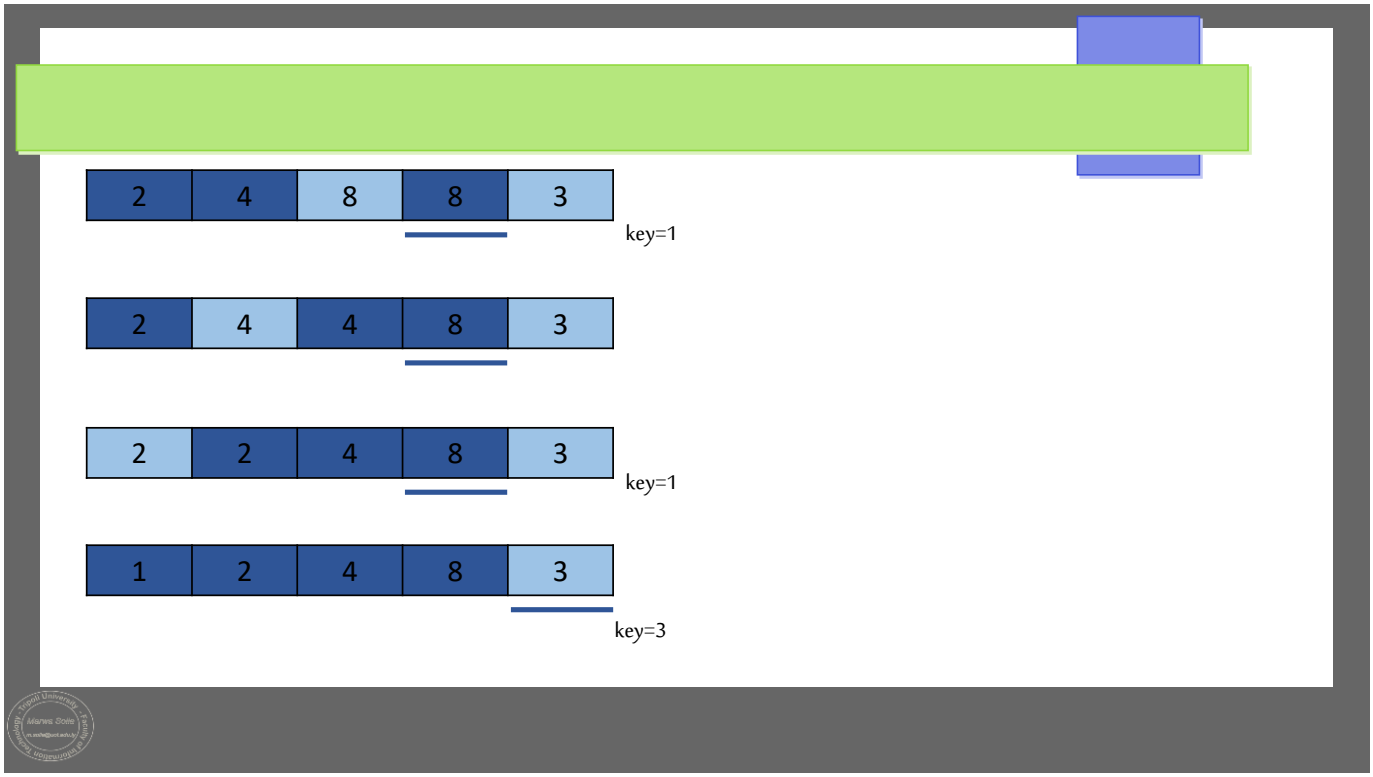


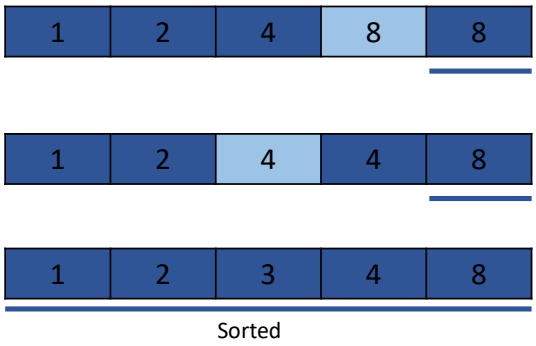
key=2



key=2


2 8 4 1 3
Sorted
UnSorted
2 8 4 1 3
2 8 8 1 3
2 4 8 1 3
key=1


2 4 8 8 3
key=1
2 4 4 8 3
2 2 4 8 3
key=1
1 2 4 8 3
key=3



Time Complexity

For i=2 to n {	n
Key= A [i]	n-1
j=i-1	n-1
While (j>0 & A [j] > key) {	$\sum_{i=2}^{n} ti$
A [j+1] = A [j]	$\sum_{i=2}^{n} (ti - 1)$
j=j-1	$\sum_{i=2}^{n} (ti - 1)$
}	
A [j+1] = key	n-1
}	

where t *is the number of while tests*

Time Complexity

Statement	Time	Best case	Worst case
For i = 2 to n {	n	n	n
Key = A[i]	n-1	n-1	n-1
j= i - 1	n-1	n-1	n-1
While (j > 0) and (A[j] > key)	$\sum_{i=2}^{n}(ti)$	n-1	n(n+1)/2 -1
A[j+1] = A[j]	$\sum_{i=2}^{n}(ti-1)$	$\sum_{i=2}^{n}(1-1)$ = 0	n(n-1)/2
j = j – 1	$\sum_{i=2}^{n}(ti-1)$	$\sum_{i=2}^{n}(1-1)$ = 0	n(n-1)/2
A[j+1] = key	n-1	n-1	n-1

Best case running time:

T(n) = n + (n-1) + (n-1) + (n-1) + 0 + 0 + (n-1)
= 5n – 4

T(n) = O (n)

Worst case running time:

T(n) = n + (n-1) + (n-1) + (n(n+1)/2 – 1) + n(n-1)/n + n(n-1)/2 + (n-1)

T(n) = O (n^2)

Example 2:

8	2	4	9	3	6

8	8	4	9	3	6

2	8	4	9	3	6

2	8	8	9	3	6

2	4	8	9	3	6

2	4	8	9	3	6

2	4	8	9	9	6

2	4	8	8	9	6

2	4	4	8	9	6

2	3	4	8	9	6

2	3	4	8	9	9

2	3	4	8	8	9

2	3	4	6	8	9

The End .